

## Doomsday Engine - Bug #869

### [Doom] E1M7 Comp Station column stuck

2010-03-03 19:48 - mfnickster

<b>Status:</b> Closed	<b>Start date:</b> 2010-03-03
<b>Priority:</b> Urgent	<b>% Done:</b> 100%
<b>Assignee:</b> skyjake	
<b>Category:</b>	
<b>Target version:</b> 1.9.10	
<b>Description</b> There is a column that is supposed to raise/lower when you cross the sector threshold from the stairs. It's in a room near the lower left of the map. The column makes a noise like it's moving but stays up. If you can't ride this column up, there's no way to get to the secret areas in the sludge pits.  Running Beta Doomsday 1.9.0-beta6.8 on a MacBook Pro with Leopard 10.5.8, using Ultimate Doom DOOM.WAD.	
<b>Labels:</b> Gameplay	
<b>Related issues:</b>	
Related to Bug #1661: [Deathkings] Bad behavior of object in lift (mobj hitbo...	<b>New</b> <b>2013-11-11</b>

#### History

##### #1 - 2010-03-04 10:45 - sonicdoommario

I have produced this all 3 times I have tested this out (twice on my desktop and once on my laptop) to confirm this too.

##### #2 - 2010-03-04 20:32 - danij

Clearly when I attempted to replicate this last time around I must have been killing all monsters beforehand.

The reason the platform does not lower is because when it tries to it thinks the move is blocked by one particular Sergeant on it (index [#11](#) - the top right of corner of the bounding box is precisely on linedef index [#491](#)).

Clearly this is not an issue in vanilla DOOM as the lift works. So, somewhere along the line we are rounding a calculation the "wrong" way (maybe the point on line side calculation?)

##### #3 - 2010-03-08 02:44 - rolandoftheeld

This bug also occurs in E4M4: Unruly Evil. There is a lowering wall with three Pinky Demons atop it that does not lower when its trigger is activated. Using the "kill" command or noclipping into the wall to kill them will allow the wall to lower.

##### #4 - 2010-03-09 17:53 - skyjake

*(originally posted by anonymous SF.net user)*

The result of P\_BoxOnLineSide() is pretty much up to chance (i.e. rounding) if one of the corners is exactly on the line. The vanilla Doom engine used fixed point mathematics for all calculations while deng uses floating point. This is probably the reason why the Sergeant was considered to be entirely on one side of the line by the original game while deng considers one of its corners to touch the neighbor sector. An easy workaround might be to minimally shrink the thing's bounding box before calling PIT\_CheckLine(). I don't know if this could have any ill side-effects, though.

Of course it's a bit strange in the first place that Height Clipping is performed when moving a floor down (or a ceiling up). If anything the move will improve the situation for the affected things, i.e. increase the available headroom.

##### #5 - 2010-08-24 22:25 - danij

Yes, I'm aware of the cause of this problem. I just haven't investigated a solution yet.

The problem with a load-time special case exception to the object's bbox is that all maps will be affected. Some, even within the original IWADs depend on the old calculation to decide their initial "stuck" state and I can see no alternative but do honour the original calculation (at least in a compatibility mode).

Similarly with the plane move height test, its counter-intuitive but DOOM depends on it. A lot of user made maps depend on its peculiar behaviour.

##### #6 - 2010-08-25 00:42 - skyjake

*(originally posted by anonymous SF.net user)*

You mean there are maps that intentionally place enemies with one corner of their bounding box touching a line and that expect these to be stuck? If there are any those maps their behavior is currently undefined when playing them in deng. Shrinking the bounding box by a very small degree, e.g. by subtracting 0.1 from the object radius when filling the bbox for PIT\_LineCheck(), would push the calculations for these corner cases towards the "safer" default, i.e. considering the object to be free.

I agree that honoring the original calculations would be a much solution, but this would require to convert the engine back to fixed point arithmetic. I don't know how much effort would be needed to do so.

**#7 - 2010-08-25 13:15 - skyjake**

*(originally posted by anonymous SF.net user)*

Ugh, missing word. That should have been "honoring the original calculations would be a much *better* solution".

**#8 - 2010-08-25 13:18 - danij**

Though such maps might not have been constructed that way intentionally, they certainly do rely on the old stuck calculation/behaviour.

I don't think its necessary to take such drastic action as to revert the whole map collision system to fixed point math. This would be my very last course of action.

A better solution would be a preprocessing step done during map conversion which actively looks for Things in this corner stuck situation. It would either flags them for special case handling or perhaps even move them out of the geometry they are stuck in.

**#9 - 2010-08-25 19:13 - skyjake**

*(originally posted by anonymous SF.net user)*

I've done a little more research:

- Unlike I suspected the behavior of this particular map is not undeterministic in original Doom: If a point is exactly on a line the P\_PointOnLineSide() function returns 1, meaning that the point is on the left side. Since most lines surrounding the moving pillar are defined counter-clockwise the left sides are on the inside of the sector and the enemies are considered to not touch the outer sector. If I flip any of the linedefs touched by an enemy the pillar is stuck using the original calculations, too.

It should be possible to reproduce the same effect using floating point calculations by comparing with a very small epsilon instead of zero in P\_PointOnLineSide().

- prboom works around the issue by disregarding the height-clipping result when lowering floors. The original code is used only when the floor movement compatibility mode is enabled, e.g. for playing demos.

**#10 - 2010-08-25 21:27 - skyjake**

*(originally posted by anonymous SF.net user)*

I've successfully tested my theory: I've changed the check in P\_PointOnLineSide() to this:

```
return ((IY - y) * IDX - (IX - x) * IDY >= 0.000001);
```

Now, if a point is very close to the line it is considered to be on the left side. Using this change the pillar is free to move even in UV (four Sergeants).

**#11 - 2010-08-25 21:49 - danij**

Good work, this is a much better solution as this test should use an epsilon in any case.

**#12 - 2010-08-25 21:51 - danij**

However its not just this that uses P\_PointOnLineSide so we'll need to check their behaviour isn't adversely affected.

**#13 - 2010-08-25 21:52 - skyjake**

*(originally posted by anonymous SF.net user)*

BTW, my change does not fix the issue mentioned in Ronald's comment (E4M4): Those demons are clearly larger than the surrounding sector, so it's not surprising that they prevent the floor from lowering. What is surprising is that (a) the level works fine with the original engine and (b) deng treats the rightmost demon (the only one present in easy skill) different than the other two. Needs more investigation.

**#14 - 2010-08-25 22:58 - danij**

Yes obviously that won't change but that is the desired outcome for that particular test.

**#15 - 2010-08-25 23:10 - skyjake**

*(originally posted by anonymous SF.net user)*

I now found out why E4M4 works in the original engine while the floor is stuck in deng: The difference is caused by the order in which the lines are

checked for collision with the map objects: The original code always iterates the lines in blockmap order. Linedef #280 (the solid wall behind the demons) is always checked before linedef #287 (the border to the moving sector). Since the demons are partially stuck in the back wall PIT\_CheckLine() will return false and the iterator will stop before checking line #287, leaving tmfloorz unmodified.

Doomsday uses a different iterator implementation which checks #287 before #280 (except for the rightmost demon). Therefore tmFloorZ will be set to the height of the moving sector, reducing the available headroom to zero.

I hope I'm making sense. Should I file a separate bug report for this issue?

#### #16 - 2010-08-27 21:38 - danij

Which branch/tag/version/revision of the source are you working with? I've recently (since the beta6.9 release) made changes to the tune of your findings when addressing a related issue in TNT.

#### #17 - 2010-08-27 23:37 - skyjake

*(originally posted by anonymous SF.net user)*

I've been running all my tests with beta6.9. I now checked out the latest git version, and the behavior is exactly the same. For reference, i.e. when I'm talking about the "original code", I'm using prboom 2.5.0 with floor motion compatibility enabled.

#### #18 - 2010-08-28 23:27 - danij

It seems this is the change I was thinking of:

<http://deng.git.sourceforge.net/git/gitweb.cgi?p=deng/deng;a=commitdiff;h=23c1997664796699db5c8121d118a6e0047a3ff9>

Clearly though that's not what we're looking at with this.

#### #19 - 2010-08-29 18:14 - skyjake

The reason why the column doesn't lower is because the game thinks the mobs on the raised floor are touching the walls they are next to. The edges of the mobj bounding boxes of the mobs do indeed touch the walls (no intersection).

This was briefly fixed by the FRACEPSILON kludge in 1.9.7 unstable builds, but a better solution is needed.

#### #20 - 2011-11-04 17:57 - danij

I have encountered this myself (only a couple of days previous in fact) however since then I have been unable to replicate the problem in subsequent play throughs.

#### #21 - 2012-03-02 11:52 - skyjake

Raised to maximum priority as the issue has wider-ranging implications for playsim accuracy/compatibility.

#### #22 - 2012-03-02 13:46 - skyjake

Offline comments from danij:

"It seems to me that you were on the right track with it being a precision issue, however I think it's the blockmap generation algorithm which needs the epsilons.

Given that the algorithm is now logically the same as id's original yet the mobs are getting stuck because their corner intercepts a linedef - it seems to me that the original algorithm placed said linedefs into blocks that wouldn't be touched due to the fixed point precision."

#### #23 - 2012-04-18 05:52 - vermil

This won't be a technical comment, but on HeXen Death Kings, Map 35, there appears to be a similar case to the Doom E1M7 column, with an Ettin and a pillar that functions as a lift (sector 15 and thing 257).

The corner of the Ettin's hitbox slightly overlaps the lift, pretty much identically to the Sargeants on column in Doom. However he can move away freely, if alerted.

However, if the player tries to use the lift without alerting the Ettin, the Ettin suddenly jumps up to the lift's higher floor and the lift becomes stuck, as if Dday is confused about what sector the Ettin was in, when the player went to use the lift.

#### #24 - 2012-05-14 16:40 - skyjake

Summarizing a little: these are the potential solutions to this issue:

1. Adjust blockmap so that the lines won't touch the mobs in question (applying 16.16 precision), suggested by [u/danij]; **OR**
2. Don't check for blocking mobs when moving a plane downwards (as running out of space is impossible and crushing will never occur), suggested by [u/inguin]. Needs a compatibility option?

#### #25 - 2012-08-30 07:15 - vermil

Both options are needed.

1. For Vanilla compat
2. For Boom compat

Regarding 1.: I actually once made a simple Vanilla Heretic boss level that relied on the fact that the behavior was 1, to make a two stage boss battle involving Maulotaurs and then an on foot D'sparil.

**#26 - 2012-08-30 09:07 - daniij**

The logically correct solution to this problem is to use the original BLOCKMAP on game-side for collision testing.

**#27 - 2012-08-30 14:17 - skyjake**

- **assigned\_to**: Jaakko Keränen

**#28 - 2012-11-12 09:13 - skyjake**

- **status**: open --> fixed  
- **milestone**: v1.9.0-beta6 --> v1.9.10

**#29 - 2012-11-16 08:21 - skyjake**

This has been addressed by applying fixed-point math to bounding box vs. line collision tests. It makes the behavior mimic that of the original DOOM, which used fixed-point math.

The blockmap iteration order had no effect on this particular issue. I examined this in the "vanilla-blockmap" branch: using the original blockmap data and iteration algorithm, the column remained stuck.

**#30 - 2012-11-16 08:21 - vermil**

May I request this report be re-opened?

While the E1M7 Column in Doom1 and platform on Sump in HeXen DK have been fixed, this case I posted above on 14/5 hasn't been fixed:

'On HeXen Death Kings, Map 35, there appears to be a similar case to the Doom E1M7 column, with an Ettin and a pillar that functions as a lift (sector 15 and thing 257).

The corner of the Ettin's hitbox slightly overlaps the lift, pretty much identically to the Sargeants on column in Doom. However he can move away freely, if alerted.

However, if the player tries to use the lift without alerting the Ettin, the Ettin suddenly jumps up to the lift's higher floor and the lift becomes stuck, as if Dday is confused about what sector the Ettin was in, when the player went to use the lift'.

While this is on a deathmatch map (HeXen DK includes some DM specific maps), it's still indicative of an issue.

**#31 - 2012-12-19 09:34 - skyjake**

jumps up to the lift's higher floor

That sounds like a different issue. Please submit a new bug report, as this one is about the E1M7 stuck column.